

REMARKS

Claims 45 and 46 (originally numbered as 47 erroneously, and renumbered in Office Action) have been canceled without prejudice.

Claims 1, 3, 17, 19, 33, and 35 are amended.

Claims 1-44 are pending.

Rejections under 35 U.S.C. §102(e)

Although not completely clear from the Office Action (due to possible typographical error), it appears that Claims 1-3, 10, 17, 26, 33, and 42 stand rejected under 35 U.S.C. §102(e) as being anticipated by *Kitagawa et al.* (U.S. Patent No. 6,078,541). If this assumption is incorrect, please restate the rejection(s).

Applicants traverse these rejections for at least the following exemplary reasons, and respectfully request that the rejections be reconsidered and withdrawn.

Kitagawa et al. disclose a system having non-volatile memory in which a loader program, decompression program and compressed device operating program are stored. With the loader and decompression programs, the device operating program is decompressed and loaded.

Independent Claim 1 is directed to a method that includes receiving file system data, storing the file system data in a plurality of reserved sectors within a non-volatile memory, compressing the file system data stored within the plurality of reserved sectors to create a compressed data block, and storing the compressed data block in at least one physical subsector within the non-volatile memory, wherein the physical subsector is associated with at least one virtual

1 sector identifiable through sector allocation information stored in a volatile
2 memory that is operatively accessible by an operating system. **Claims 2, 3 and 10**
3 each depend from independent Claim 3 and recite additional claim limitations.

4 *Kitagawa et al.* neither disclose nor suggest a method and storing this type
5 of compressed data block in at least one physical subsector within the non-volatile
6 memory, and having the physical subsector associated with at least one virtual
7 sector identifiable through sector allocation information stored in a volatile
8 memory that is operatively accessible by an operating system.

9 Independent **Claim 17** is directed to a computer-readable medium having
10 computer-executable instructions for performing steps that include receiving file
11 system data, storing the file system data in a plurality of reserved sectors within a
12 non-volatile memory, compressing the file system data stored within in the
13 plurality of reserved sectors to create a compressed data block, and storing the
14 compressed data block in at least one physical subsector within the non-volatile
15 memory, wherein the physical subsector is associated with at least one virtual
16 sector identifiable through sector allocation information stored in a volatile
17 memory that is operatively accessible by an operating system. **Claim 26** depends
18 from independent Claim 17 and recites additional claim limitations.

19 Again, *Kitagawa et al.* neither disclose nor suggest a computer readable
20 medium for performing storage of this type of compressed data block in at least
21 one physical subsector within the non-volatile memory, and having the physical
22 subsector associated with at least one virtual sector identifiable through sector
23 allocation information stored in a volatile memory that is operatively accessible by
24 an operating system.

25

1 Independent Claim 33 is directed to an arrangement that includes an
2 operating system and a device driver. As recited, the operating system is
3 configured to exchange input/output (I/O) requests with the application and
4 exchange corresponding file system requests with the device driver, and the device
5 driver is configured to store the file system data received from the operating
6 system in a plurality of reserved sectors within the non-volatile memory, compress
7 the file system data stored within in the plurality of reserved sectors to create a
8 compressed data block, and store the compressed data block in at least one
9 physical subsector within the non-volatile memory. The physical subsector is
10 associated with at least one virtual sector identifiable through sector allocation
11 information stored in a volatile memory that is operatively accessible by the
12 operating system. Claim 42 depends from independent Claim 33 and recites
13 additional claim limitations.

14 Clearly, *Kitagawa et al.* neither disclose nor suggest such an arrangement.
15 There simply are not physical subsectors associated with virtual sectors in
16 *Kitagawa et al.* and/or related sector allocation information stored in volatile
17 memory and used by the operating system of the device taught by *Kitagawa et al.*

18 Consequently, all of the pending claims, Claims 1-44, are each patentable
19 over *Kitagawa et al.*

20

21 **Rejections under 35 U.S.C. §103(a)**

22 Claims 2, 3, 7-16, 18, 19, 23-32, 34, 35, and 39-44 stand rejected under 35
23 U.S.C. §103(a) as being unpatentable over *Kitagawa et al.* in view of *Franaszek et*
24 *al.* (U.S. Patent No. 6,401,181).

1 Applicants traverse these rejections for at least the following exemplary
2 reasons, and respectfully request that the rejections be reconsidered and
3 withdrawn.

4 As mentioned above, *Kitagawa et al.* disclose a system having non-volatile
5 memory in which a loader program, decompression program and compressed
6 device operating program are stored.

7 *Franaszek et al.* disclose techniques for dynamically allocating physical
8 memory and recovering unused memory. These techniques include using a
9 translation table having pointers that map CPU generated memory addresses to
10 actual sectors of the physical memory. Memory controllers are employed to help
11 control memory pools and populate free lists of pointers for memory space that is
12 available for use. The techniques disclosed in *Franaszek et al.* are for volatile
13 dynamically allocated memory systems. The techniques of *Franaszek et al.* do not
14 address compressed file systems. The techniques taught by *Franaszek et al.* do not
15 therefore disclose or otherwise suggest methods and arrangements that have a
16 compressed data block stored in at least one physical subsector within the non-
17 volatile memory, and having the physical subsector associated with at least one
18 virtual sector identifiable through sector allocation information stored in a volatile
19 memory that is operatively accessible by an operating system.

20 Furthermore, there appears no motivation to combine the teachings of
21 *Kitagawa et al.* and *Franaszek et al.* that would result in the claimed invention,
22 especially since *Kitagawa et al.* are interested in simply decompressing a
23 compressed operating system from non-volatile memory and thereafter operating a
24 file system strictly in volatile memory and *Franaszek et al.* are interested in

1 managing dynamically allocated volatile memory and do not address compression
2 or non-volatile based file systems.

3 Turning to the claims, **Claims 2, 3, and 7-16** depend from independent
4 Claim 1, which is directed to a method that includes receiving file system data,
5 storing the file system data in a plurality of reserved sectors within a non-volatile
6 memory, compressing the file system data stored within in the plurality of reserved
7 sectors to create a compressed data block, and storing the compressed data block in
8 at least one physical subsector within the non-volatile memory, wherein the
9 physical subsector is associated with at least one virtual sector identifiable through
10 sector allocation information stored in a volatile memory that is operatively
11 accessible by an operating system.

12 Neither *Kitagawa et al.* and/or *Franaszek et al.* disclose or reasonably
13 suggest a method and storing this type of compressed data block in at least one
14 physical subsector within the non-volatile memory, and having the physical
15 subsector associated with at least one virtual sector identifiable through sector
16 allocation information stored in a volatile memory that is operatively accessible by
17 an operating system.

18 **Claims 18, 19 and 23-32** depend from independent **Claim 17**, which is
19 directed to a computer-readable medium having computer-executable instructions
20 for performing steps that include receiving file system data, storing the file system
21 data in a plurality of reserved sectors within a non-volatile memory, compressing
22 the file system data stored within in the plurality of reserved sectors to create a
23 compressed data block, and storing the compressed data block in at least one
24 physical subsector within the non-volatile memory, wherein the physical subsector
25 is associated with at least one virtual sector identifiable through sector allocation

1 information stored in a volatile memory that is operatively accessible by an
2 operating system.

3 Neither *Kitagawa et al.* and/or *Franašek et al.* disclose or suggest a
4 computer readable medium for performing storage of this type of compressed data
5 block in at least one physical subsector within the non-volatile memory, and
6 having the physical subsector associated with at least one virtual sector identifiable
7 through sector allocation information stored in a volatile memory that is
8 operatively accessible by an operating system.

9 **Claims 34, 35 and 39-44** depend from independent **Claim 33**, which is
10 directed to an arrangement that includes an operating system and a device driver.
11 As recited, the operating system is configured to exchange input/output (I/O)
12 requests with the application and exchange corresponding file system requests with
13 the device driver, and the device driver is configured to store the file system data
14 received from the operating system in a plurality of reserved sectors within the
15 non-volatile memory, compress the file system data stored within in the plurality of
16 reserved sectors to create a compressed data block, and store the compressed data
17 block in at least one physical subsector within the non-volatile memory. The
18 physical subsector is associated with at least one virtual sector identifiable through
19 sector allocation information stored in a volatile memory that is operatively
20 accessible by the operating system. **Claim 42** depends from independent **Claim 33**
21 and recites additional claim limitations.

22 *Kitagawa et al.* and/or *Franašek et al.* fail to disclose or suggest such an
23 arrangement.

24 Regarding **Claims 4-6, 20-22 and 36-38**, the Office Action correctly stated
25 that these dependent claims are patentable over *Kitagawa et al.* and/or *Franašek*

1 et al. The amendments to independent claims 1, 17 and 33 further distinguish the
2 pending claims over these references and all of the cited art.

3 Consequently, all of the pending claims, Claims 1-44, are each patentable
4 over the cited art.

5 **Conclusion**

6 The pending claims have been placed in condition for allowance and are
7 clearly patentable over the cited art and should therefore be allowed.

8
9 Respectfully Submitted,

10 Date: 1/27/2004

11 By: T. A. Jolly

12 Thomas A. Jolly
13 Reg. No. 39,241

14
15
16
17
18
19
20
21
22
23
24
25